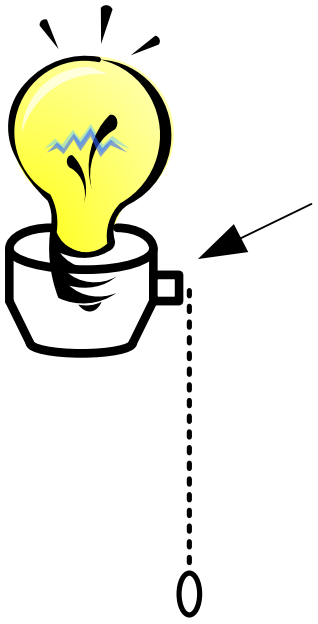
UM EECS 270 F22
Introduction to Logic Design

9. Latches & Flip-Flops

- **Combinational Circuits**: Output is a function of only the *current* inputs
  - So far, we've been looking at combinational circuits
- **Sequential Circuits:** Output is a function of both the *current and previous* inputs
  - Example: Pull light switch

Inputs happen know
Inputs happen before



pull = 1 → switch light from off to on or on to off

Result of pull = 1 depends on whether the light is currently on or off, which in turn depends on the entire history of "pulls"

output : inputs + state of the circuit

light_next = f(pull, light_current)

pull = 0 → light_next = light_current

pull = 1 → light_next = $\overline{\text{light\_current}}$

light_current is called the **state** of the circuit. Note that it is actually the state that is dependent on all previous inputs

Finite State Machines (aka Finite Transition Systems)

it is a mathematical model that can describe logical circuits

- Mathematical model of *sequential* behavior
- Applications: hardware, software, protocols, etc.
- Specified by two functions:

the value of x in the next time. : is a function of state now, input now.

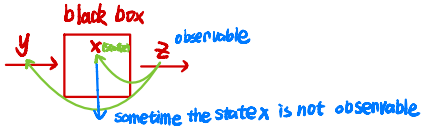
state input

$$\text{Next State: } X^+ = f(X,Y)$$

$$\text{Output: } Z = g(X) \text{ or } g(X,Y)$$

where

- X and X^+ are current and next *state* variables
- Y are *input* variables
- Z are *output* variables



- A *state* is a complete assignment to (valuation of) the state variables X
- Next State function specifies the *transitions* between states
- Output function specifies the *observable* outputs of the state machine

That's a mathematical concept of finite transition system or a finite state machine

Finite State Machine (FSM) Example

In this example, we only have one state variable X.

State Transition Equation

$x \in [0,9]$

$y \in [1,2]$

$x^+ = (x > y)?\, x - y : y$

state variable

$x \in [0,9]$

input

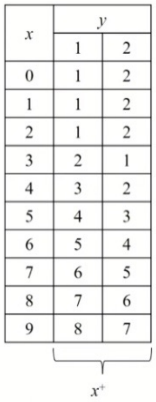
$y \in [1,2]$

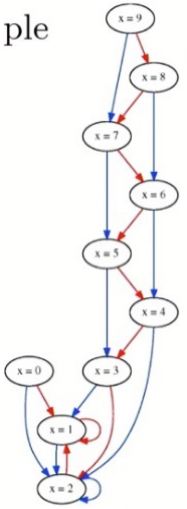
$x^+ = (x > y)?\, x - y : y$

State Transition Table

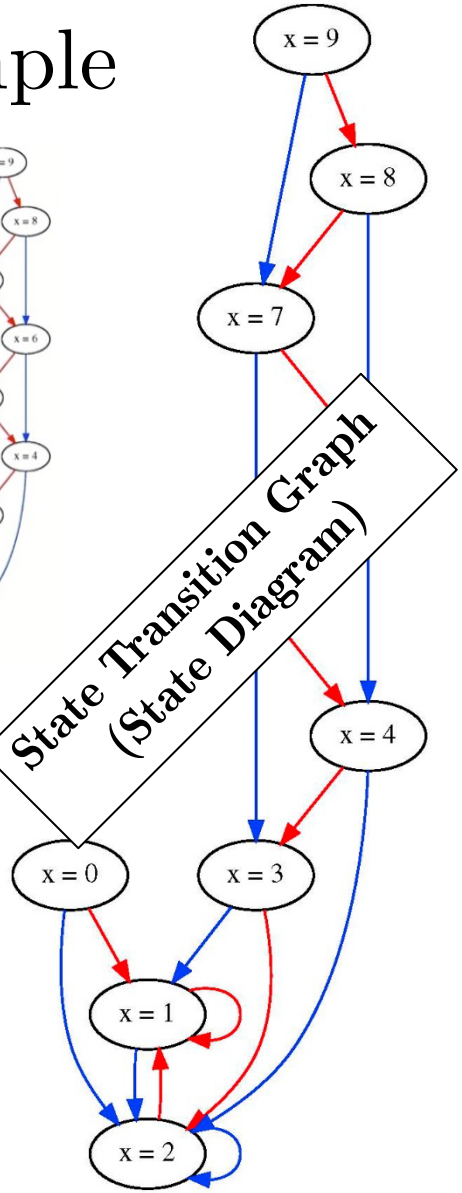
x	y = 1	y = 2
0	1	2
1	1	2
2	1	2
3	2	1
4	3	2
5	4	3
6	5	4
7	6	5
8	7	6
9	8	7

x^+

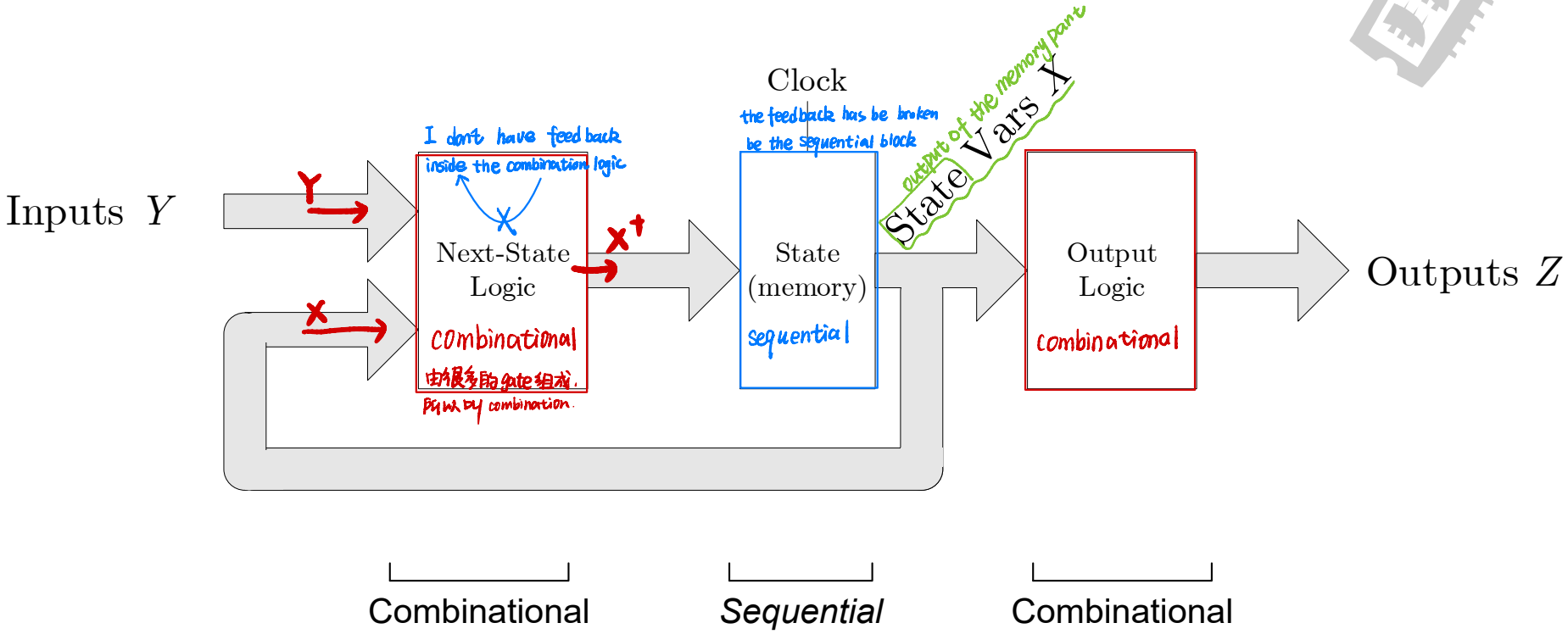




State Transition Graph (State Diagram)

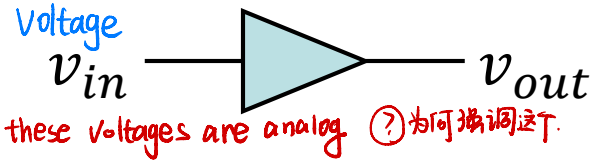


High-Level Structure of Sequential Circuits



First we'll cover the building blocks of sequential logic design…

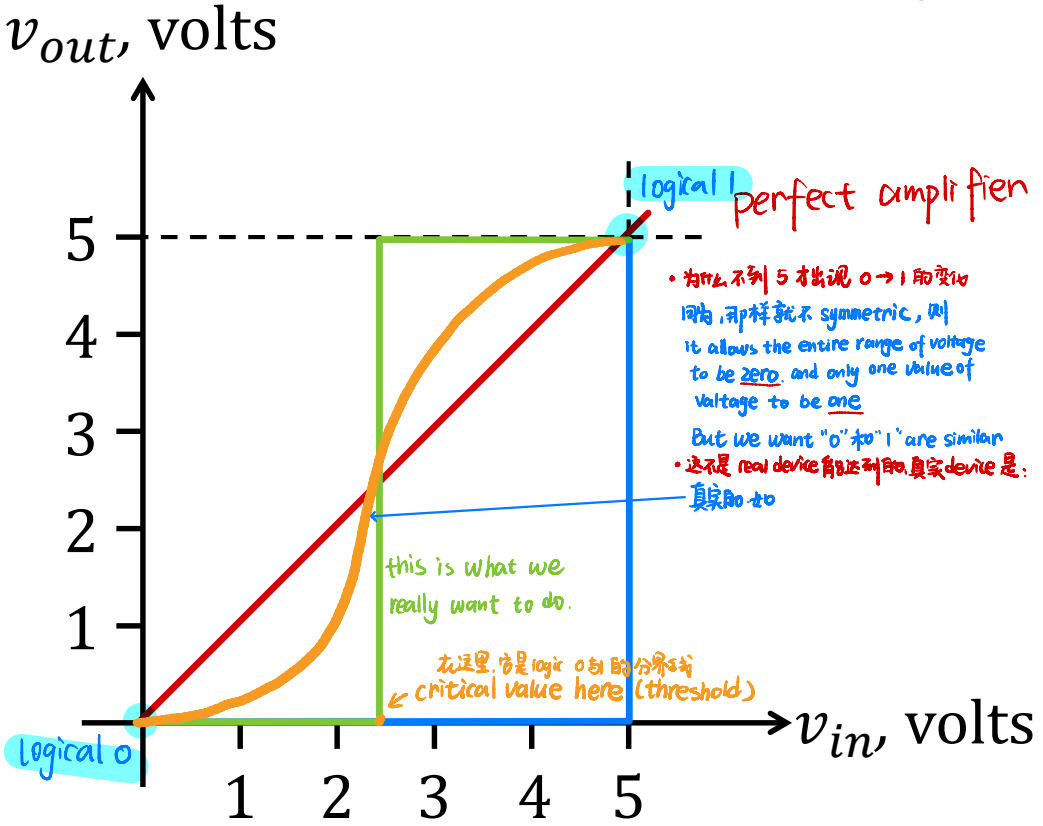
Is the World Really Digital?



these voltages are analog ?为何强调这个.

v_{in}	v_{out}
0	0
1	1

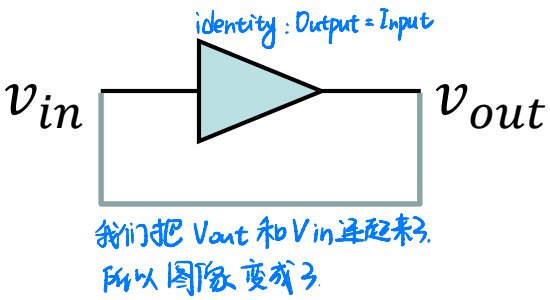
我们称其为switch device的原因是：(ideal的)
当Vinput在0~2.5时，output一直是0.
而当Vinput超过2.5到2.5~5时，output瞬间变成1

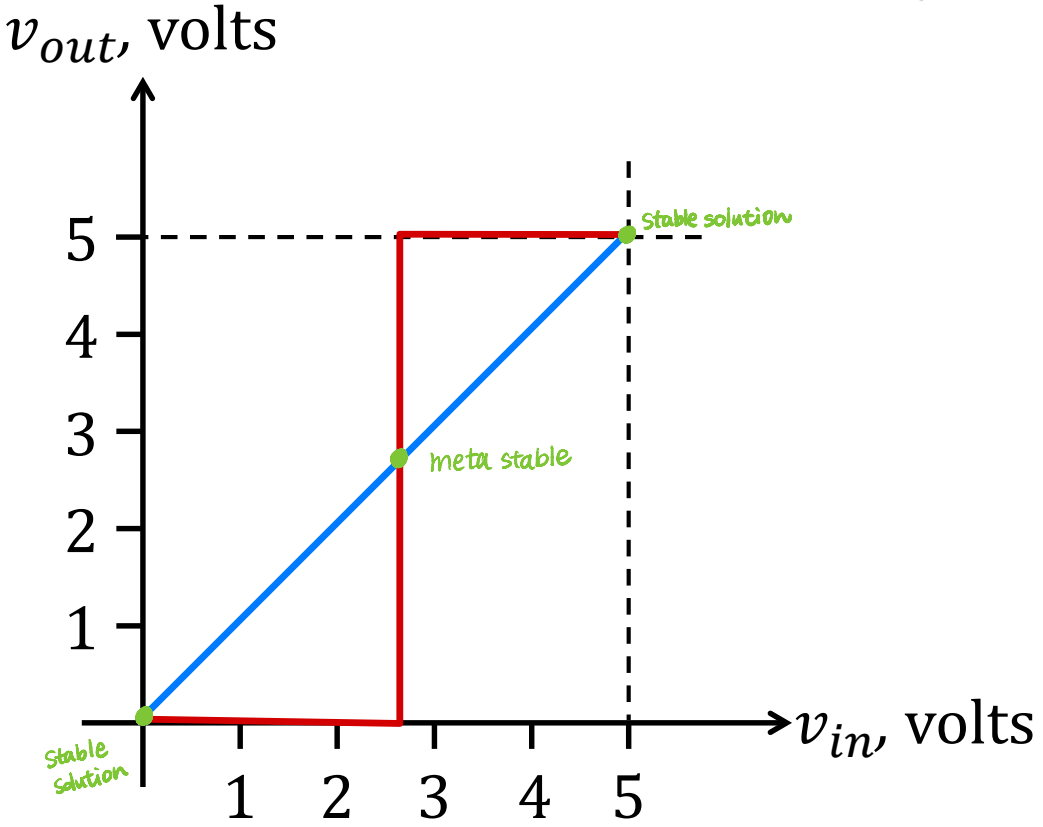


Is this a good digital device?

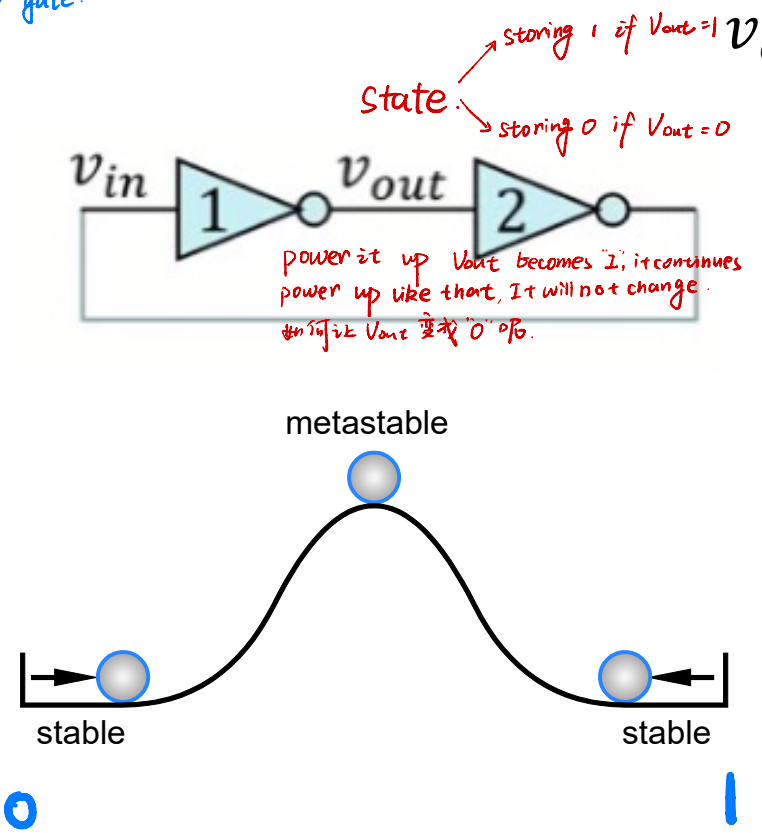
No.

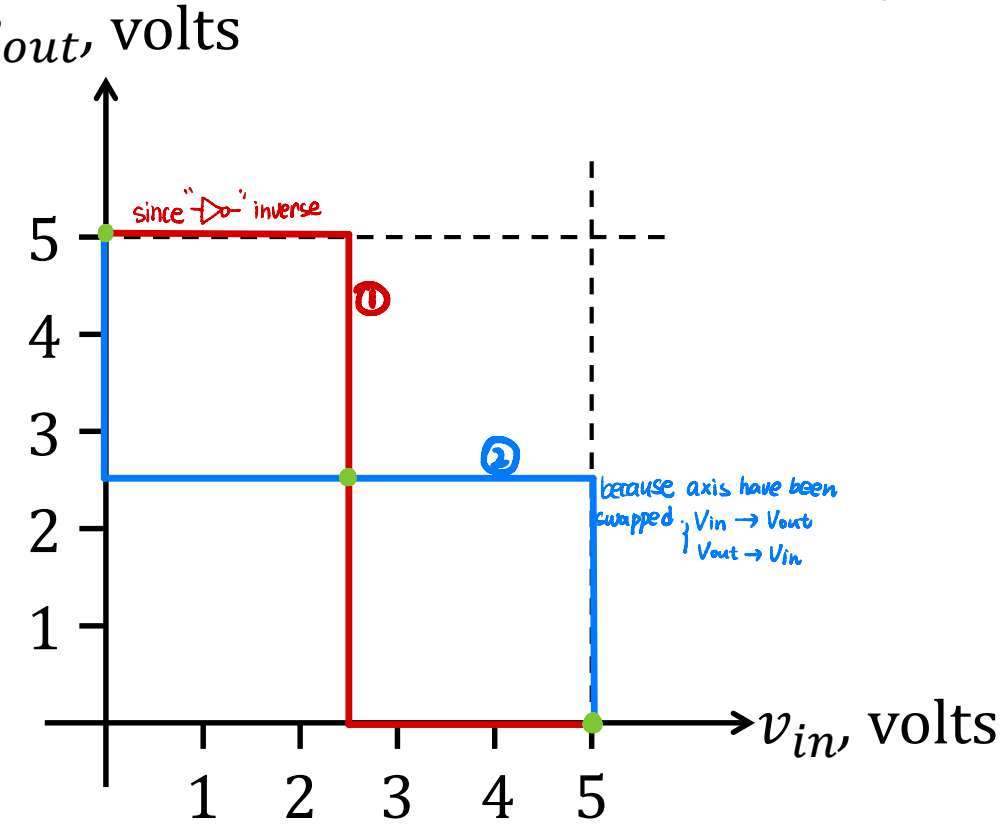
Basic Memory Cell: Feedback



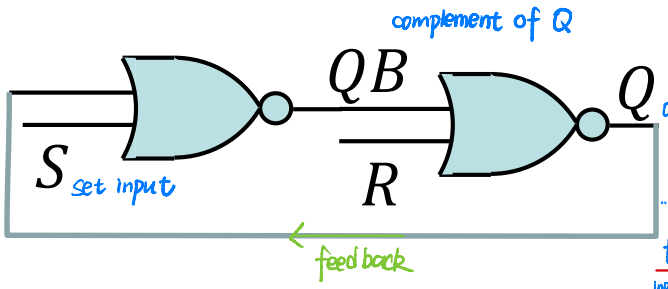


Bistable Element





SR (Set-Reset) Latch



"1" is a control value for "or" & "nor"

input "0" 和 "1" reset the latch to be "0" 和 "1"

这种情况说明的原因是 feedback：

让R→1，使得Q→0，然后等等直到Q上的值从 feedback loop中转回来，然后使得QB→1 之后我们就 能remove R了

做一下 push，让R是1一段时间

How long？：the delay of both gate.

然后 feedback action 完成 不管输入是啥，现在是"0"或"1"

independent

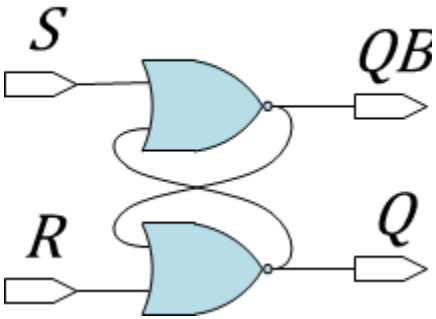
S	R	Q^+ next value of Q	QB^+ next value of QB	Function
0	0	Q	QB	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Invalid

自锁住

都受外层input影响

输出达成这三种 action

我们输出2bits 的信息：S 和 R

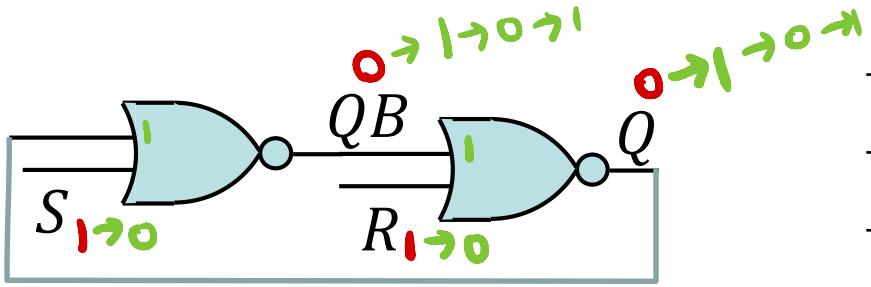


two "nor" gate in a loop

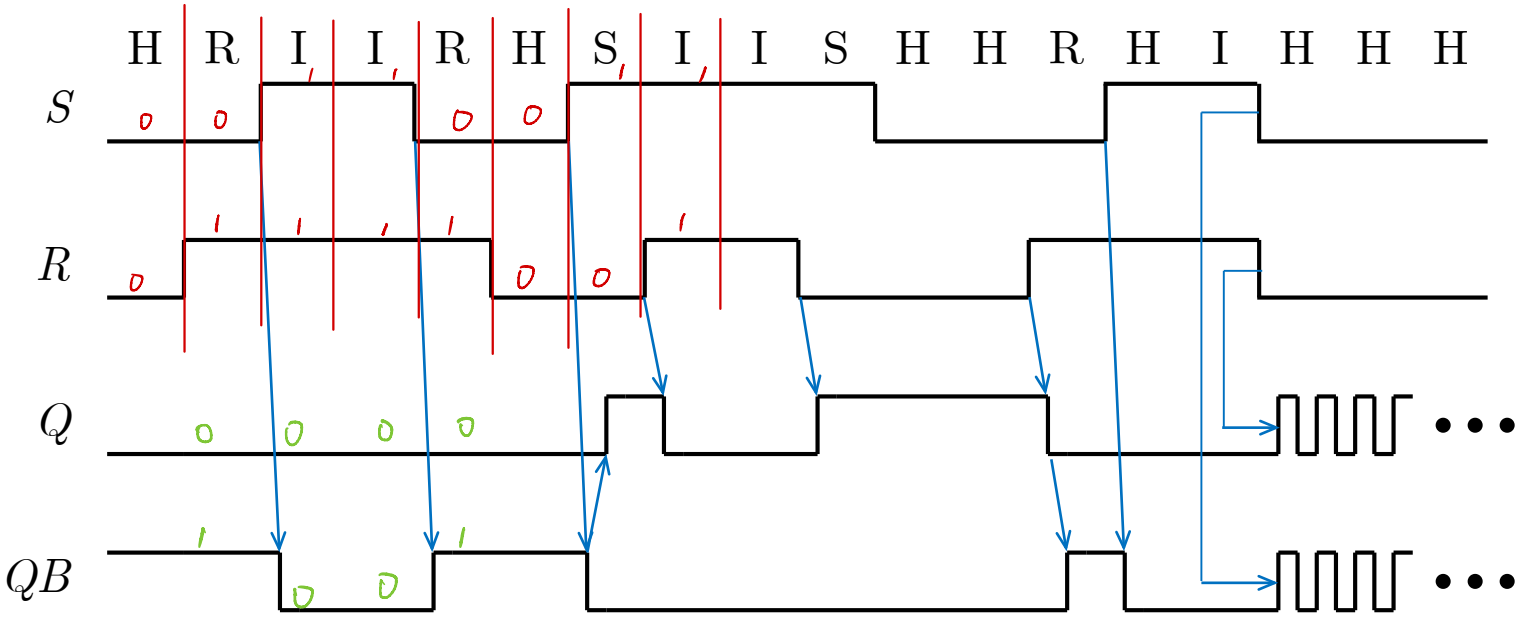
是我们所说的：set/reset latch

Cross-Coupled NOR Gates

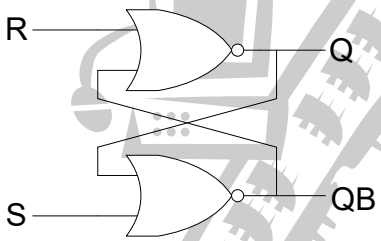
SR Latch Timing



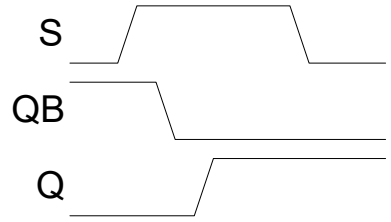
S	R	Q^+	QB^+	Function
0	0	Q	QB	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	Invalid



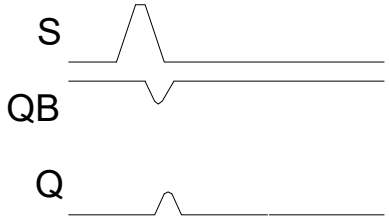
Metastability



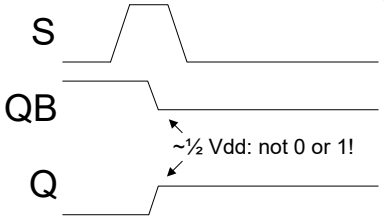
- SR Latch behavior is sensitive to S (or R) pulsewidth



S pulse width is long enough: OK

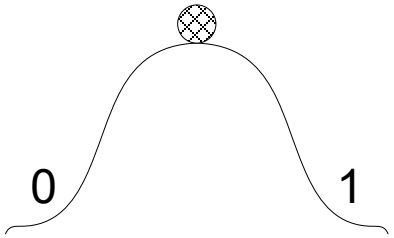


S pulse width is too short:
no change registered!



S pulse width is *almost* long enough: metastability!

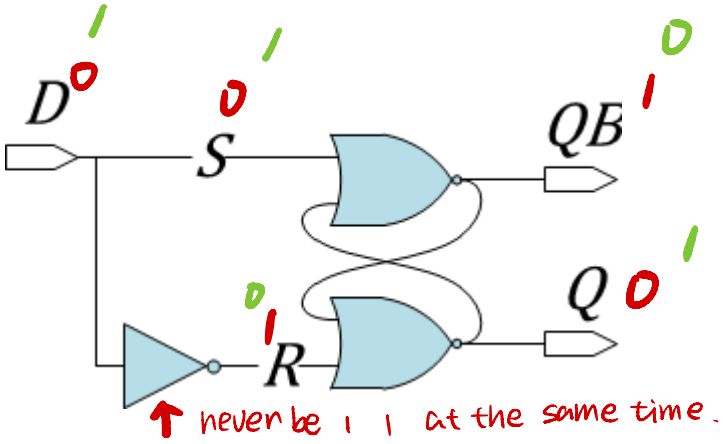
- A **metastable** state is one in which Q and QB reach an unstable equilibrium voltage *that is neither 0 or 1*
  - The voltages will eventually and non-deterministically return to 0 or 1



- Therefore, latch manufactures specify a *minimum pulse width* on the inputs that must be satisfied for the device to properly change state
  - Transitioning from the SR = 11 state to the SR = 00 state may also cause metastability

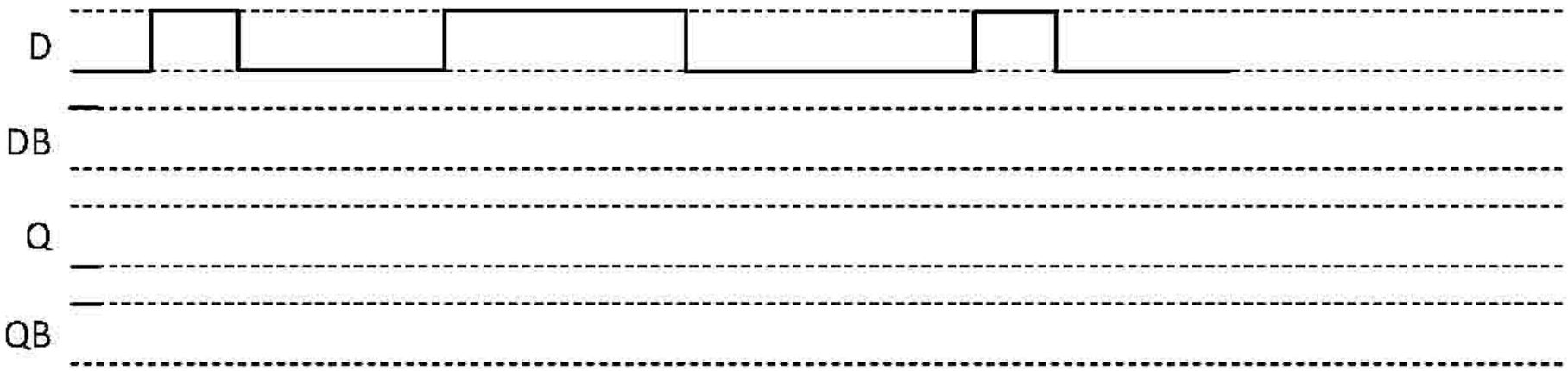
ModelSim Demo
Latch Timing / Metastability

D (Data or Delay) Latch



D	Q^+	QB^+	Function
0	0	1	Reset
1	1	0	Set

$Q^+ = D$

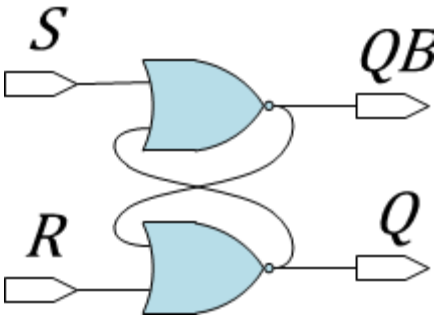


the state will change each time per-cycle.

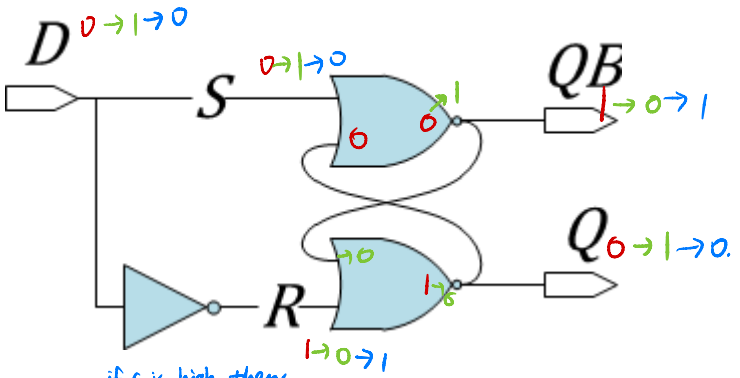
Latch Names and Symbols

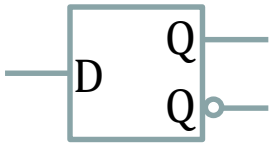
set reset latch.

SR Latch



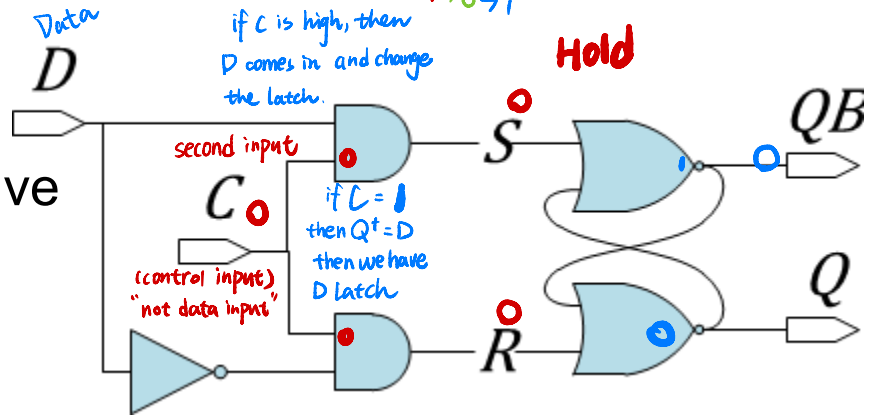
D Latch

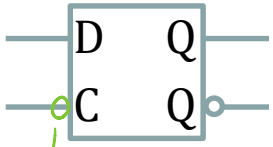




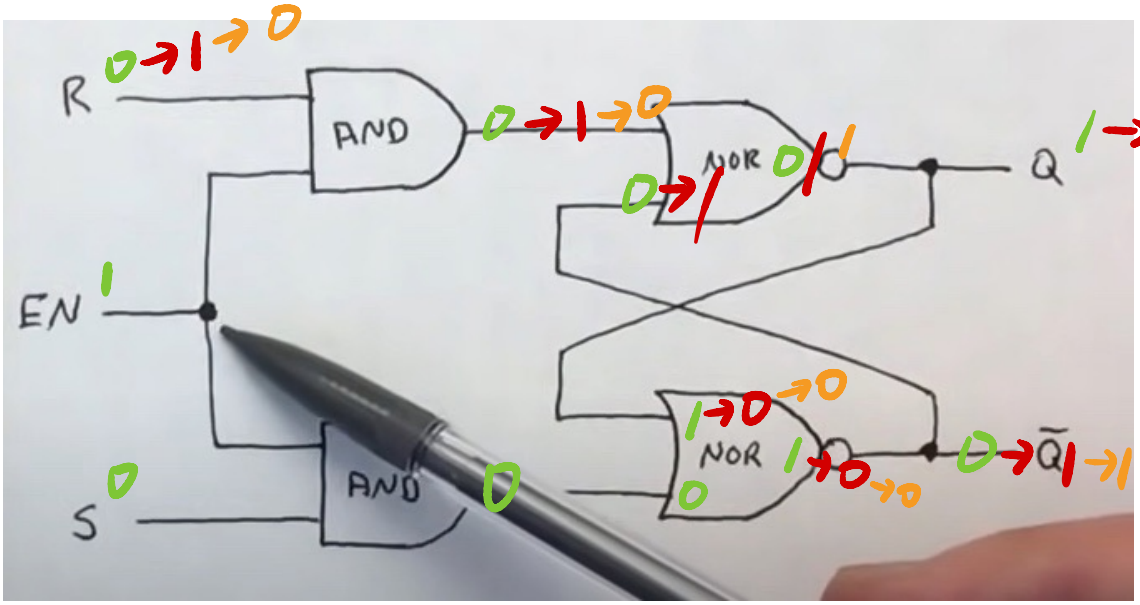
D是来set或reset整个系统.
而C是控制.D是否传入的.

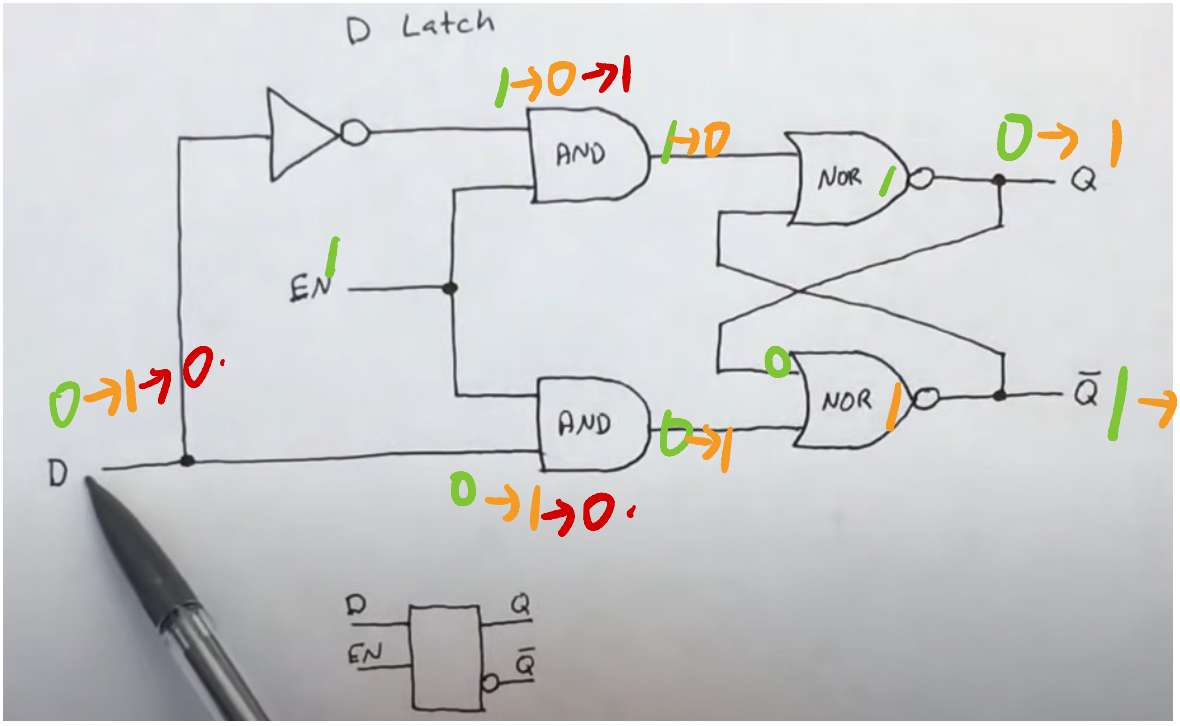
(Positive) Level Sensitive D Latch

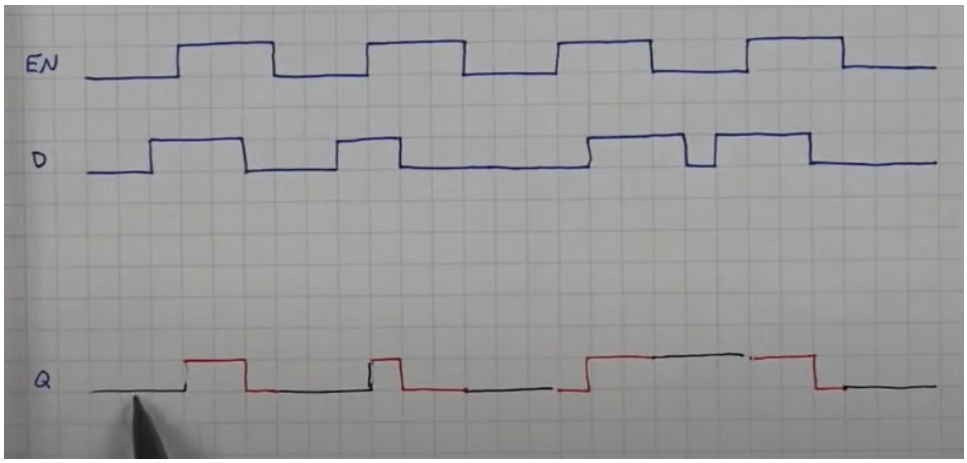




有时候还会有一个"0"表明C responds to the zero level of clock instead of "1"

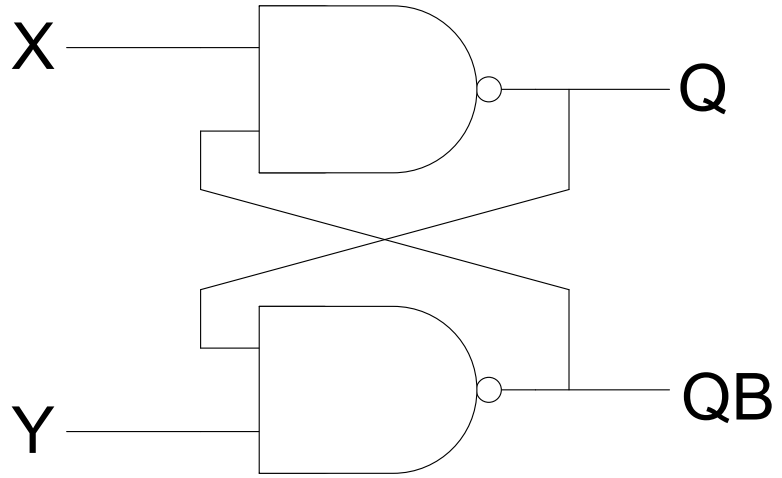

R
AND
NOR
Q
EN
AND
NOR
S


D Latch
AND
NOR
Q
EN
AND
NOR
D
D
EN
Q


EN
D
Q

In-Class Exercise

Complete the truth table for the SR-like latch below.
What are appropriate names for X and Y?

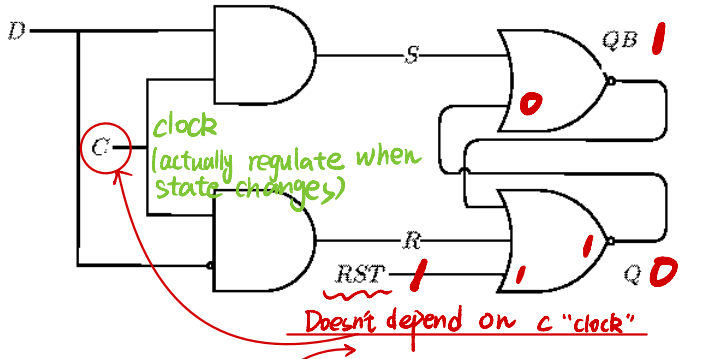


X	Y	Q^+	QB^+
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	QB

$X \rightarrow S\_L$; $Y \rightarrow R\_L$

D Latch Initialization

usually some signal called an initialization signal or reset signal.
make everything go to "0"

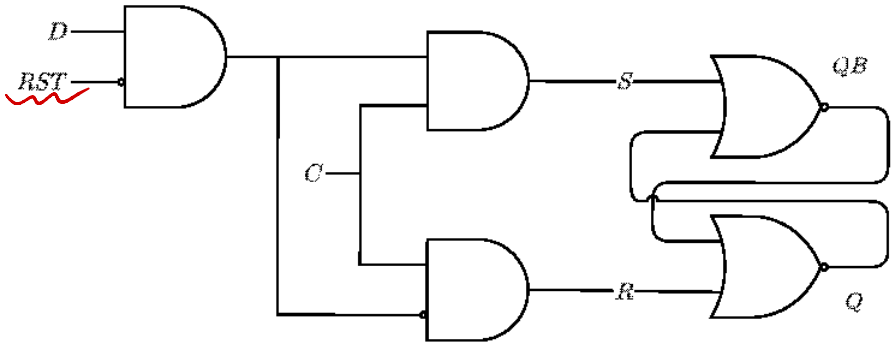


How to reset clock D latch.

名字来源 (异步)

Asynchronous Reset
(Initialize to 0)

It overrides the clock (don't depend on clock).

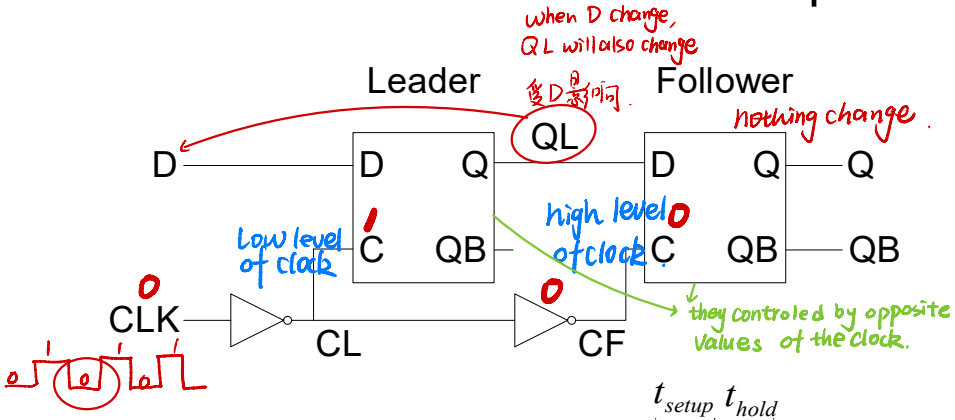


Synchronous Reset
(Initialize to 0)

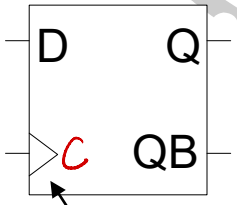
synchronous with the clock.

Edge-Triggered D Flip-flop

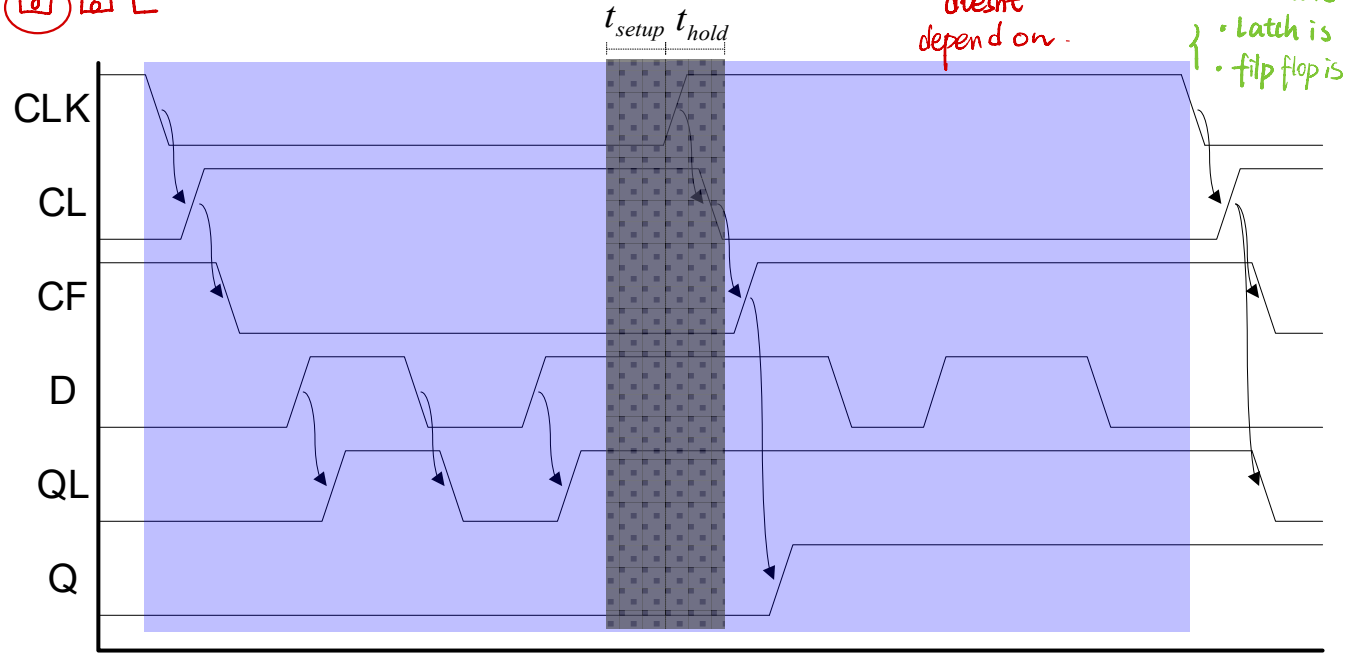
- Want a device that will sample its inputs *only once per clock cycle*



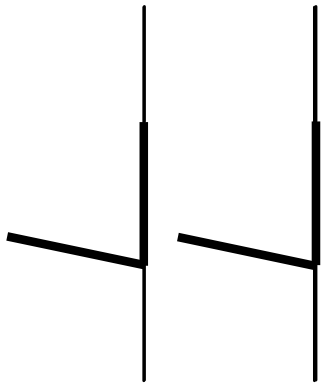
D	CLK	Q^+	QB^+
0	↑	0	1
1	↑	1	0
X	0	Last Q	Last QB
X	1	Last Q	Last QB



Indicates edge-trigged behavior



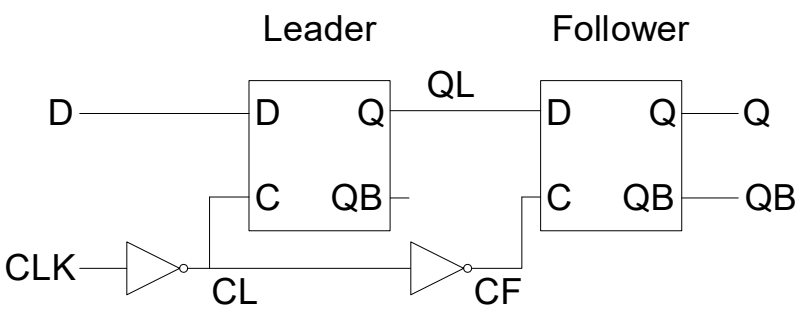
Leader transparent *Follower transparent*



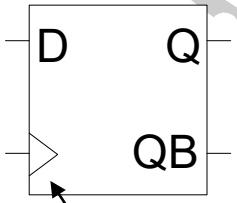
"Double-door" analogy

Edge-Triggered D Flip-flop

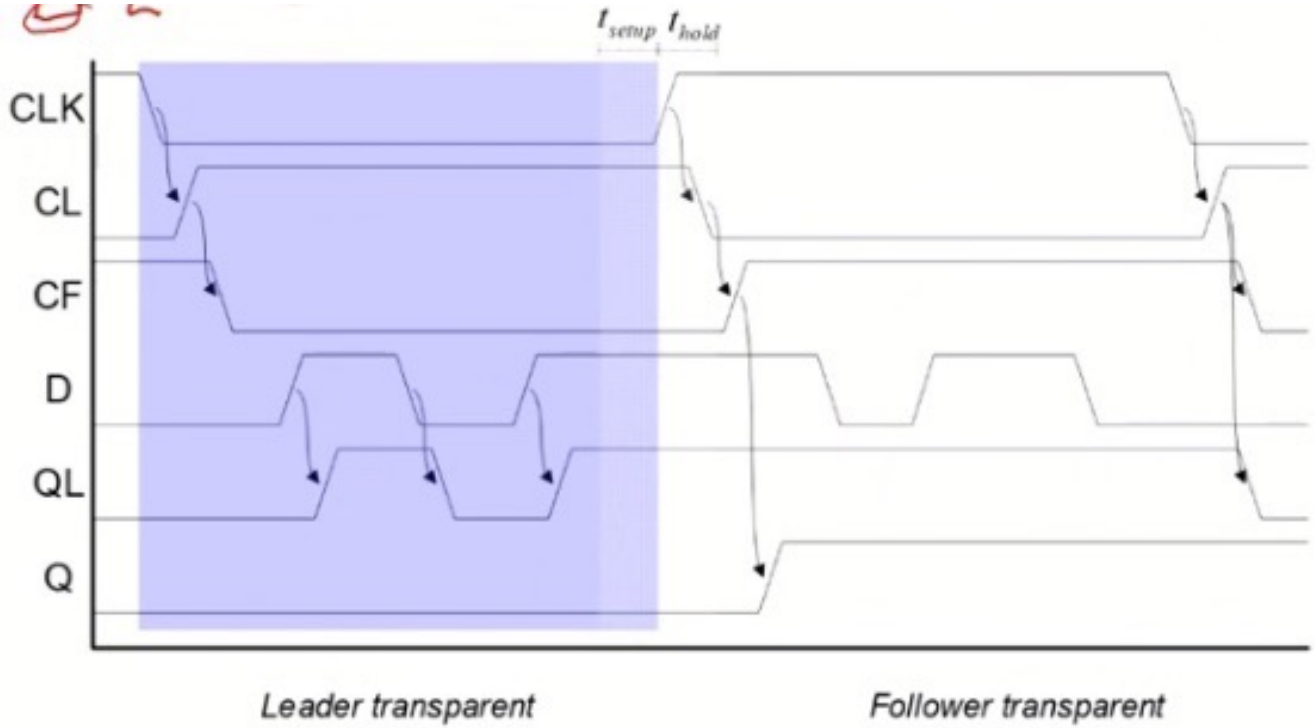
- Want a device that will sample its inputs *only once per clock cycle*

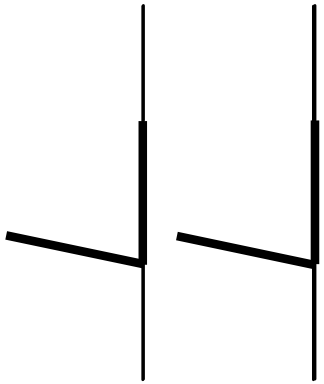


D	CLK	Q	QB
0	↑	0	1
1	↑	1	0
x	0	Last Q	Last QB
x	1	Last Q	Last QB

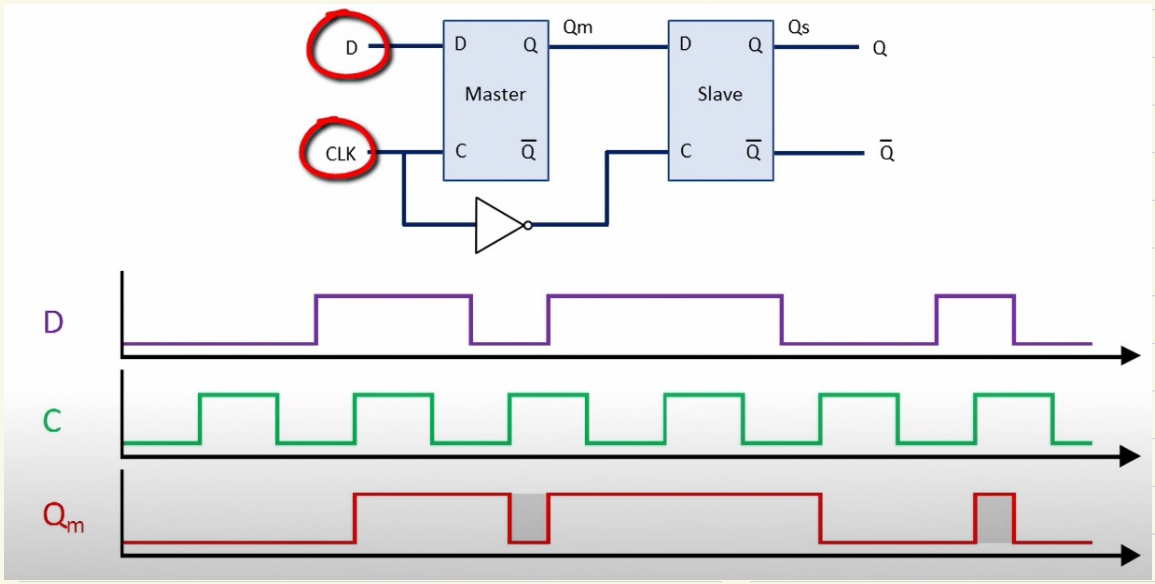


Indicates edge-trigged behavior





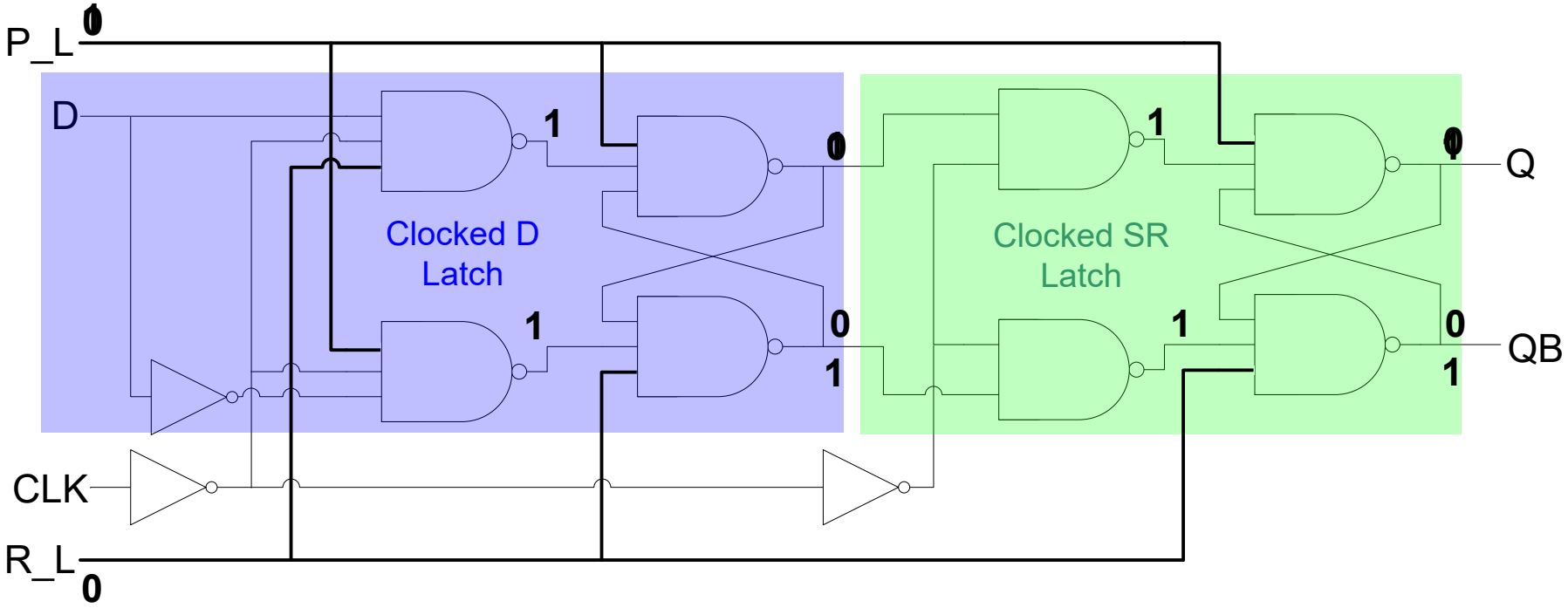
"Double-door" analogy


D
CLK
D
Q
Master
C
Q̄
Qm
D
Q
Slave
C
Q̄
Qs
Q
Q̄
D
C
Qm

Variations on the D Flip-Flop

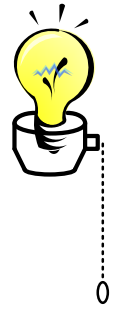
- **Asynchronous inputs** force a flip-flop to a specified state when asserted, *independent of the clock signal*
  - **Asynchronous preset**: sets the output of the flip-flop to 1 *independent of the clock*
  - **Asynchronous reset**: resets the output of the flip-flop to 0 *independent of the clock*

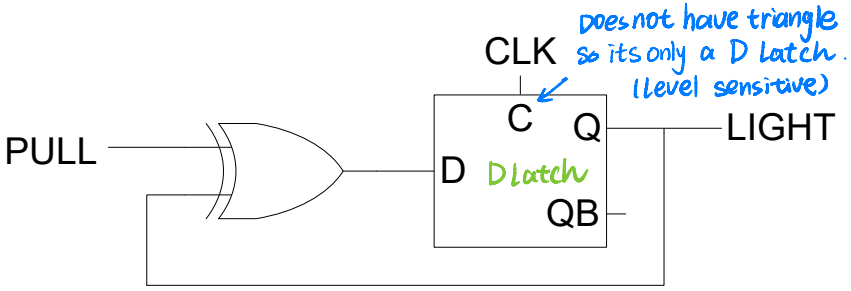
D Flip-Flop with Asynchronous Reset & Preset



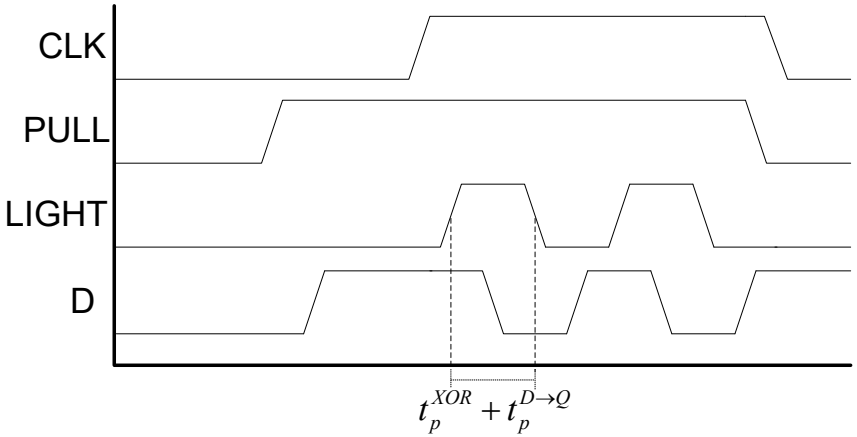
D Latch Application

- Pull light switch revisited - how to implement using a D latch?



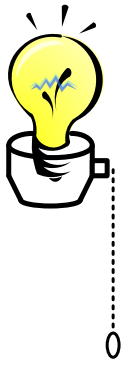


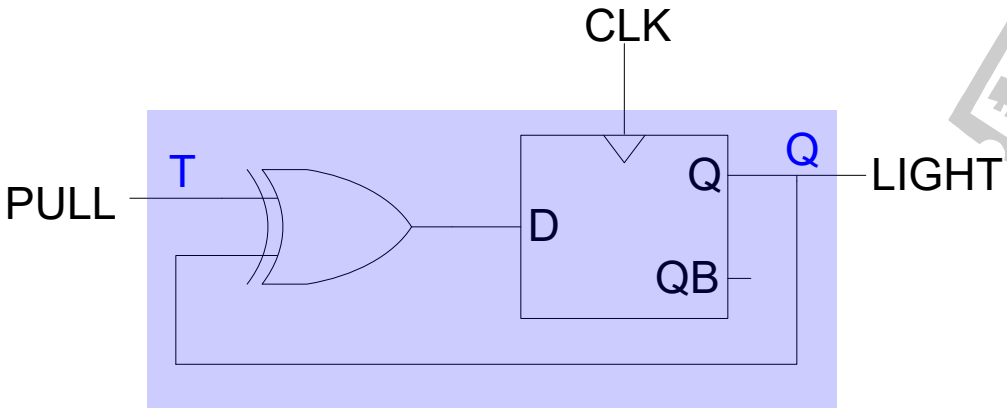
- Problems with this implementation?
  - What if pull is held high while the latch is transparent??



We've made a strobe light!

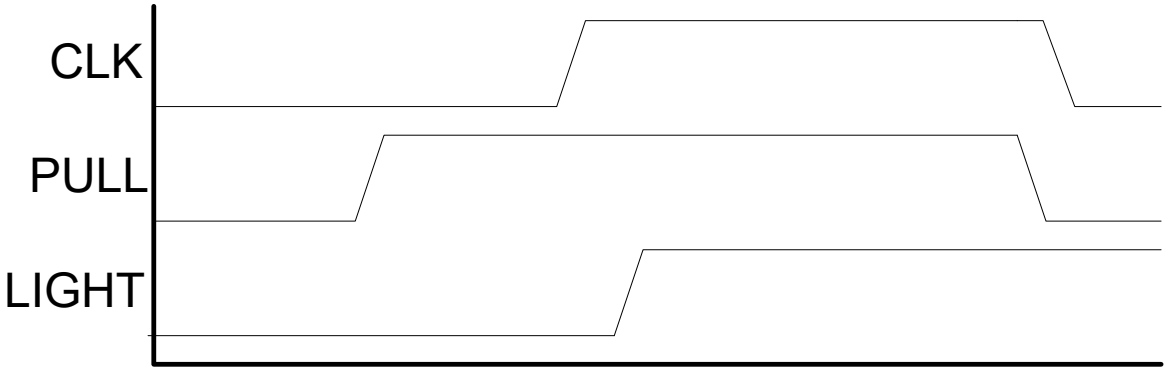
- Pull light switch revisited: D flip-flop implementation

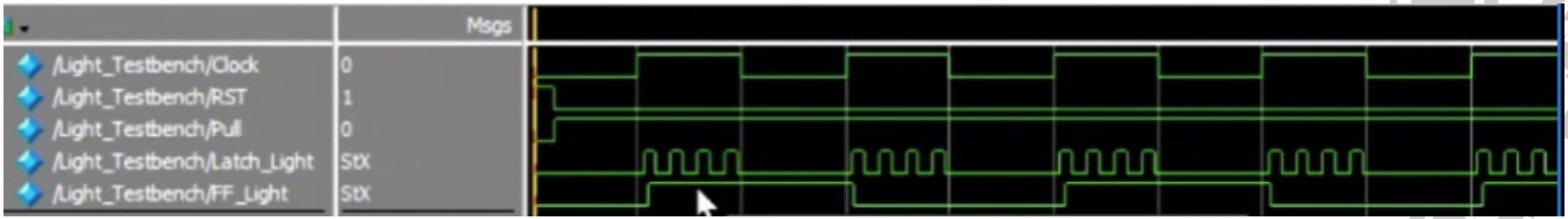




*This device is known as a **toggle flip-flop***

- Only one light toggle per clock cycle!





ModelSim Demo
Latch vs FF Timing
Pull Light Switch